فهرست:

مقدمه:

اولين قدم در بسياري از الگوريتم هاي پردازش تصوير و بينايي ماشين، حذف نويز از تصوير مي باشد، چرا كه بدون حذف نويز اين الگوريتم ها نتايج خوبي توليد نمي كنند . براي مثال نويز نقطه اي كه به مانند لكه هاي سياه يا روشن روي تصوير مي باشد يكي از انواع شايع نويزها مي باشد . تا به حال فيلترهاي مختلفي براي كاهش اين نوع نويز ارائه شده است . در گروه اول، فيلترهايي ارائه شد كه عملكرد خوبي در حذف نويز داشته ولي عموماً تصوير را مات و جزئيات آن را از بين مي بردند . گروه دوم فيلترهايي كه ارائه شد، جزئيات تصوير را حفظ كرده ولي ميزان كاهش نويز آنها كم بود و گروه سوم فيلترها، آنهايي هستند كه تلاش مي كنند نويز را حذف كرده در حالي كه لبه ها و جزئيات تصوير را نيز حفظ كنند . در بسياري از موارد اين فيلترها عملكرد خوبي دارد ولي چنانچه جزئيات ريز تصوير كم باشد و يا سطوح خاكستري موجود در همسايگي پيكسل ها اختلاف زيادي با هم داشته باشند، باعث مات شدن تصوير مي گردد.

امروزه با گسترش روز افزون روش های مختلف اخذ اطلاعات گسسته مانند پویشگرها و دوربین های دیجیتالی، پردازش تصویر کاربرد فراوانی یافته است. تصاویر حاصله از این اطلاعات همواره کم و بیش همراه مقداری نویز بوده و در مواردی نیز دارای مشکل محوشدگی مرزهای نمونه های داخل تصویر می باشند که موجب کاهش وضوح تصویر دریافتی می گردند. مجموعه عملیات و روش هایی که به منظور کاهش عیوب و افزایش کیفیت ظاهری تصویر مورد استفاده قرار میگیرد، پردازش تصویر نامیده می شود. اگرچه حوزه های کار با تصویر بسیار وسیع است ولی عموماً محدوده مورد توجه در چهار زمینه ی بهبود کیفیت ظاهری(Enhancement)، بازسازی تصاویر مختل شده(Restoration)، فشرده گی و رمزگذاری تصویر (Compression and Coding) و درک تصویر توسط ماشین (Understanding) متمرکز می گردد.

بهبود تصاویر شامل روش هایی مثل استفاده از فیلتر محو کننده و افزایش تضاد برای بهتر کردن کیفیت دیداری تصاویر و اطمینان از نمایش درست آن ها در محیط مقصد است. بینایی ماشین به روش هایی می پردازد که به کمک آن ها می توان معنی و محتوای تصاویر را درک کرد تا از آن ها در کارهایی چون رباتیک و محور تصاویر استفاده شود. پردازش تصویر از هر دو جنبه نظری و عملی پیشرفت های چشمگیری داشته است و بسیاری از علوم به آن وابسته اند.

کاربردهای پردازش تصویر

زمینه های مختلف کاربرد پردازش تصویر عبارتند از صنعت، هواشناسی، شهرسازی، کشاورزی، علوم نظامی و امنیتی، نجوم و فضا نوردی، پزشکی، فناوری های علمی، باستان شناسی، تبلیغات، سینما، اقتصاد، روانشناسی و زمین شناسی که در ادامه درباره هر کدام مختصراً بحث شده است.

صنعت

امروزه کمتر کارخانه پیشرفته ای وجود دارد که بخشی از خط تولید آن توسط برنامه های هوشمند بینایی ماشین کنترل نشود. خطای بسیار کم، سرعت زیاد، هزینه نگهداری بسیار پایین، عدم نیاز به حضور اپراتور 24 ساعته و خیلی مزایای دیگر باعث شده که صنایع و کارخانه ها به سرعت به سمت پردازش تصویر و بینایی ماشین روی بیاورند. دستگاهی ساخته شده که قادر است کیک های پخته را از کیک هایی که نیاز به پخت مجدد دارند، تشخیص دهد و آنها را به صورت اتوماتیک به بسته بندی بفرستد و کیک هایی که نیاز به پخت دارند را دوباره برای پختن ارسال کند.

یکی دیگر از دلایل استفاده از بینایی ماشین قابلیت دیدن و اندازه گیری محصولاتی است که دیدن یا اندازه گیری آنها با چشم غیر مسلح غیر ممکن است. عناصر تشکیل دهنده یک سیستم بینایی ماشین نرم افزار هوشمند بینایی است که ورودی خود را از دوربین های نصب شده در بخش های مختلف خط تولید می گیرد و بر اساس تصاویر دریافتی دستورات لازم برای کنترل ماشین های صنعتی را صادر می کند. پردازش تصویر در تشخیص دمای کوره هایی که هیچ وسیله ی مکانیکی و الکترونیکی تحمل دمای آنها را ندارد، کاربرد دارد. دوربین های حرارتی می توانند مشکل بخشی از سازه ی مورد نظر را تشخیص دهند.

هواشناسی

از آنجایی که در علم هواشناسی تشخیص و پیش بینی آب و هوا اکثراً از طریق تصاویر هوایی و ماهواره ای انجام می گیرد، پردازش تصویر در این علم کاربرد زیادی دارد و دقت و سرعت پیش بینی آب و هوا و طوفان ها را بسیار بالا می برد. جبهه های پرفشار، کم فشار، گردبادها و گرداب های بوجود آمده در سطح کره زمین را می توان مشاهده کرد

شهرسازی

با مقایسه عکس های مختلف از سال های مختلف یک شهر می توان میزان گسترش و پیشرفت آن را مشاهده کرد. کاربرد دیگر پردازش تصویر می تواند در کنترل ترافیک باشد. با گرفتن عکس های هوایی از زمین ترافیک هر قسمت از شهر مشخص می شود.

قبل از ساختن یک شهر می توان آن را توسط کامپیوتر شبیه سازی کرد که به صورت دو بعدی از بالا و حتی به صورت سه بعدی از دید های مختلف، یک شهرک چطور ممکن است به نظر برسد. تصاویر ماهواره ای که از شهرها گرفته می شود، می تواند توسط فیلترهای مختلف پردازش تصویر فیلتر میشود و اطلاعات مختلفی از آن استخراج شود. به طور مثال این که شهر در چه قسمت هایی دارای ساختمان ها، آب ها یا راه های بیشتری است و همین طور می توان جاده هایی که داخل یا خارج از شهر کشیده شده اند را تحلیل کرد.

کشاورزی

این علم در بخش کشاورزی معمولاً در دو حالت کاربرد دارد. یکی در پردازش تصاویر گرفته شده از ارتفاعات بالا مثلاً از هواپیما و دیگری در پردازش تصاویر نزدیک به زمین .

در تصاویر دور به عنوان مثال می توان تقسیم بندی اراضی را تحلیل کرد. همچنین می توان با مقایسه تصاویر دریافتی در زمان های متفاوت میزان صدمات احتمالی وارد به محیط زیست را دید. به عنوان مثال می توان برنامه ای نوشت که با توجه به محل رودخانه ها و نوع خاک مناطق مختلف، به صورت اتوماتیک بهترین نقاط برای کشت محصولات مختلف را تعیین می کند.

تصاویر نزدیک هم در ساخت ماشین های هرز چین اتوماتیک کاربرد دارد. امروزه ماشین های بسیار گران قیمت کشاورزی وجود دارند که می توانند علف های هرز را از گیاهان تشخیص بدهند و به صورت خودکار آنها را نابود کنند .

برای مثال یکی از پروژه های جالب در بخش کشاورزی، تشخیص خودکار گل زعفران برای جداسازی پرچم قرمز رنگ آن بوده است. این پردازش که توسط نرم افزار Stigma detection® انجام گرفته است.

علوم نظامی و امنیتی

پردازش تصویر بخصوص بینایی هوشمند، کاربردهای بسیاری را در علوم نظامی و امنیتی دارند و این کاربرد برای دولت اکثر کشورها بسیار مهم است. به عنوان مثال موشک هدایت شونده خودکاری وجود دارد که می تواند روی در یک ساختمان قفل کند و حتی می تواند به درز بین در و دیوار آن ساختمان که حساس ترین جای ساختمان است به راحتی نفوذ کند. این موشک به صورت اتوماتیک این قسمت را شناسایی کرده و به سمت آن حمله می کند.

در مسائل امنیتی هم کاربرد پردازش تصویر کاملاً در زندگی ما مشهود است. دوربین های که به صورت اتوماتیک از ماشین هایی که تخلف رانندگی انجام می دهند عکس برداری می کند .

از سیستم های امنیتی دیگر می توان سیستم تشخیص اثر انگشت اتوماتیک را نام برد. در لپ تاپ های جدید قابلیت finger print به آنها اضافه شده و می تواند صاحب لپ تاپ را توسط اثر انگشت شناسایی کند .

کد امنیتی دیگری که همیشه همراه انسان حمل می شود، چشم انسان است. دانشمندان ثابت کرده اند که پترن های (Pattern)موجود در مردمک چشم هر انسان منحصر به فرد است و هیچ دو فردی در دنیا وجود ندارند که پترن هایی که در مردمک چشم آنها وجود دارد دقیقاً مثل هم باشد. از همین روش برای شناخت افراد و سیستم های امنیتی استفاده می شود.

در کل این خواص بیومتریک در انسان بسیار زیاد است .عرض و طول صورت، فاصله بین انگشتان دست، طول و عرض انگشت ها، فاصله ی بندها از یکدیگر و حتی خط های کشیده شده کف دست و هزاران خاصیت دیگر،

تماماً خصوصیاتی هستند که برای انسان ها منحصر به فرد هستند.دوربین هایی وجود دارند که به صورت دید در شب، قادر هستند چیزهایی را که ما نمی بینیم، ببینند و پردازش کنند.اسلحه های خودکاری ساخته شده اند که به صورت اتوماتیک و دقیق نشانه گیری می کنند .

پردازش تصویر همینطور با پردازش تصاویر گرفته شده از فاصله های دور هم می تواند در علوم نظامی و امنیتی کمک کند.به عنوان مثال دوربینی قادر است با سرعت بسیار زیاد یک توپ را دنبال کند.این مسئله کاربرد بسیار زیادی در مسائل نظامی دارد.

نجوم و فضا نوردی

ساخت دستگاه های اتوماتیک رصد آسمان و ثبت وقایع آسمانی به صورت خودکار از کاربردهای پردازش تصویر است که امروزه روی آن کار می شود .

از پروژه های جدید در بخش نجوم که بخشی از آن توسط سیستم پردازش تصویر انجام می شود، تهیه نقشه سه بعدی از کل عالم کائنات است !

پردازش تصویر در فضانوردی هم کاربرد زیادی دارد. در تصاویر دور می توان سطح سیارات و همچنین سطح قمرها را اسکن کرده و اطلاعات بسیار ریزی از آنها استخراج کنیم .

کاربرد دیگر پردازش تصویر در فیلترکردن عکس هایی است که توسط تلسکوپ های فضایی مختلف از جمله هابل(Hubble Space Telescope) ، از فضا گرفته می شود.

کاربرد دیگر آن حذف گرد و خاک و جو سیاره ها از تصاویر به کمک تصویربرداری IR و X-RAY به صورت همزمان و ترکیب این تصاویر است .

در تصاویر نزدیک هم کاربرد دارد، از جمله هدایت مریخ نوردها، فرود فضاپیماهای بدون سرنشین و الصاق تجهیزات جدید به ایستگاههای فضایی به صورت خودکار.

از امکانات سایت گوگل، امکاناتی است به نام Google Mars که این برنامه دقیقاً مانند Google Earth عمل می کند با این تفاوت که Google Earth سطح زمین را در هر زمان که بخواهید و در هر نقطه ای از زمین و از ارتفاع های بسیار پائین هم نشان می دهد ولی Google Mars دقیقاً همین کار را برای سطح سیاره مریخ انجام می دهد .

پزشکی

یکی از مهمترین کاربردهای پردازش تصویر در علم پزشکی است. در جایی که ما نیاز داریم تمام عکس ها با نهایت شفافیت و وضوح گرفته شوند زیرا دیدن تمام جزئیات لازم است. جراحی های ریز microsurgery با ایجاد یک سوراخ کوچک و فقط دیدن محل جراحی توسط پزشک، از راه دور و توسط بازوهای رباتیک بسیار دقیق انجام می شوند

فناوری های علمی

پردازش تصویر در افزایش سرعت پیشرفت های علمی تأثیر فوق العاده داشته است. اولین و مشخص ترین تأثیر آن را می توان در علم عکاسی یا هنر دید. شکار لحظه های شگفت آوری که در کسری از ثانیه اتفاق می افتد، بالا بردن وضوح عکس های گرفته شده و ایجاد افکت های خیره کننده، از دستاوردهای پردازش تصویر است .

همچنین در توسعه تکنولوژی پیشرفته gps (Global Positioning Systems) کمک زیادی داشته و تهیه نقشه های سه بعدی از جاده ها در تمام نقاط جهان، از کاربردهای دیگر آن است. با به وجود آمدن این علم، مسابقات ربات های فوتبالیست به صورت جدی دنبال شد .

این علم در پیشرفت علوم پایه فیزیک ، شیمی و مخصوصاً تحقیقات فیزیکی و مکانیکی، کمک فراوانی کرده است. به عنوان مثال وسیله ای برای حمل و نقل کالاها در مسیرهای صعب العبور ساخته شده است. قبل از ساخت آن، رفتار چهارپایان در حالت های مختلف توسط کامپیوتر تحلیل و عیناً به دستگاه آموزش داده شده است.در کل پردازش تصاویر به علت سرعت زیاد آن، در ساخت وسایل مکانیکی پر سرعت، کاربرد زیادی دارد. وسیله ای وجود دارد که قادر است ، توپی که با سرعت بسیار زیاد به سمت پائین می آید را مهار کند .

باستان شناسی

در علم باستان شناسی تنها مدارک باقی مانده از دوران باستان، دست نوشته ها، نقاشی ها و غارنگاری های قدیمی است .تهیه تصاویر از بناهای گذشته و بازسازی مجازی این بناهای تاریخی یکی از کاربردهای پردازش تصویر در این علم است. همچنین می توان نقاشی ها و غارنگاری ها را مورد پردازش دقیق قرار داد و شکل آنها را همان طور که در ابتدا بوده اند، شبیه سازی کرد .حتی می توان مکانهای باستانی را از زوایایی که تصاویر مستندی از آنها وجود ندارد، شبیه سازی کرد.

امروزه یکی از پروژه های پر سر و صدای بازسازی بناهای باستانی، بازسازی شهر روم باستان توسط دانشمندان ایتالیایی است. هم اکنون توریست ها با زدن عینک های مخصوص می توانند در خیابان های شهر روم باستان قدم بزنند.

تبلیغات

از مقایسه تبلیغات دهه ی 70 و 80 میلادی با تبلیغات امروزی می توان تأثیر تکنولوژی را در تبلیغات کاملاً درک کرد. تغییر شکل تبلیغات از اشکال مربع و زاویه دار به شکل های دایره ای، تغییر رنگ تبلیغات و هزاران تغییر دیگر. یکی از مهمترین فاکتورهای فروش و دلایل بالا رفتن یا پایین آمدن فروش، شکل و نحوه ی بسته بندی کالاست. پردازش تصویر می تواند به ما کمک کند تا قبل از تولید یک بسته بندی آن را شبیه سازی کنیم. با ادغام کردن علم الگوریتم ژنتیک با پردازش تصویر می توان برنامه ای را نوشت که به صورت اتوماتیک به ساختن بسته بندی های مختلف بپردازد و آنهایی که از نظر کاربران زیباتر و جالب تر به نظر خواهند آمد را به ما معرفی نماید.

سینما

اولین علمی که پردازش تصویر در آن مورد استفاده قرار گرفت، هنر و سینما بود. یکی از تکنولوژی های برتر دنیا motion capture است که در آن یک کاراکتر انیمیشنی قادر است حرکات دست انسان را تقلید کند. امروزه این سیستم جهت ساخت فیلم ها و بازی های کامپیوتری مورد استفاده قرار می گیرد .
در پردازش تصویر قابلیتی به نام هیستوگرام (Histogram) وجود دارد که با آن قادرند تصاویر را شفاف یا تیره تر کرده و یا هر تغییر مورد نیاز دیگری را روی تصاویر با توجه به منحنی ها و نمودارهای هیستوگرام بدهند .
در سینما برای اینکه تصویری شفاف به نظر آید، با استفاده از یک کره ی نقره ای رنگ، تصاویر اطراف دوربین را هم ثبت می کنند. بنابراین تصویر نسبت به محیط اطراف خود شفافیت غیر قابل تصوری پیدا می کند.

اقتصاد

در دنیای امروز تمام نوآوری ها، به نوعی مستقیم یا غیر مستقیم باعث تغییراتی در اقتصاد گروهی از کشورها و یا کل دنیا می شوند .پردازش تصویر هم، به صورت مستقیم و غیر مستقیم در اقتصاد تأثیر گذار است. در تبلیغات، سیاست، فضانوردی، کشاورزی، شهرسازی، سینما، پزشکی و علوم نظامی می تواند تأثیر غیر مستقیمی در اقتصاد کشورها داشته باشد. همچنین از تأثیر مستقیم آن در اقتصاد، می توان به وجود شعبه های بانک بدون کارمند اشاره کرد. این شعبه ها قادرند به صورت خودکار سریال چک ها و قبوض پرداختی را بخوانند، نوع اسکناس ها را تشخیص دهند و تا حد زیادی از کارهای یک بانک عادی را انجام دهند .

روانشناسی

بحث تاثیر رنگ در روحیه انسان اهمیت بسیار زیادی دارد به طوری که در روانشناسی گرایشی به نام روانشناسی رنگ وجود دارد. در این علم در مورد رنگ ها و تأثیر هر یک بر روح و جسم انسان صحبت می شود. به عنوان مثال رنگ قرمز بیشتر تأثیر را در چشم انسان دارد. در حالی که رنگ سبز بیشترین تأثیر را در مغز انسان دارد. همچنین رنگ آبی باعث ایجاد حس آرامش و اطمینان در انسان می شود. به همین دلیل در سخنرانی های اکثر سیاستمداران دنیا از پرده آبی رنگ در پشت سر آن ها استفاده می شود.

با پردازش تصویر می توان به راحتی تصاویر ثابت و متحرک را ویرایش کرد. به طور مثال رنگ آبی را برای ایجاد حس اطمینان یا رنگ سبز را برای حس زیبایی و قرمز را برای ایجاد هیجان در تصاویر پر رنگ تر کرد .

زمین شناسی

با پردازش تصویر می توان کانی های مختلف را از روی رنگ و اندازه آن ها شناسایی و دسته بندی کرد. همچنین در زمین شناسی برای پی بردن به مواد تشکیل دهنده کانی ها از روش پرتونگاری tomography)) استفاده می کنند و پردازش تصویر در این بخش می تواند سرعت و دقت این روش را بسیار بالا ببرد. کاربرد دیگر آن این است که دانشمندان با مقایسه کردن ارتفاع آب در سال های مختلف، در واقع روند تند شدن یا کند شدن کاهش آب در سطح زمین را مورد بررسی قرار می دهند .

نتیجه گیری

رد پای پردازش تصویر در بسیاری از علوم و صنایع مشاهده می شود و بعضی از این کاربردها آنچنان به پردازش تصویر وابسته هستند که بدون آن، اساساً قابل استفاده نمی باشند. کاربرد پردازش تصویر در هر یک از زمینه هایی که بحث شد، بسیار گسترده است.

فصل اول:

اشنایی با محیط .netو برنامه نویسی در #c

قبل از هر چیز بهتر است تعریف دقیقی از کلمات فریم ورک یا چهار چوب داشته باشیم.

در تعریف net. میتوانیم بگوییم که:"چهار چوب net. یک پلتفرم جدید است که توسط مایکرسافت برای طراحی و توسعه نرم افزار تولید شده است.".

اگر به تعریفی که در بالا برای چارچوب NET. اوزده شده است دقت کنید .مشاهده می کنید که این تعریف محدود به نوع خاصی از برنامه ها نیست . در حقیقت در مورد نوع برنامه هایی که با NET. میتوان نوشت محدودیتی وجد ندارد که بخواهیم انرا ذکر کنیم . از چارچوب NET. می توان برای طراحی برنامه های تحت ویندوز وبرنامه های تحت وب وسرویسهای مبتنی بر وب و....استفاده کنید.

#c از دو زبان ++ Cو Java متولد شده است! حاوی بسیاری از جنبه های++ C می باشد اما ویژگی های شیء گرایی خودش را از جاوا به ارث برده است.

#C اگرچه از++ Cگرفته شده است اما یک زبان "خالص" شیء گرا (Object oriented) می باشد. هر دو زبان یاد شده جزو زبان های هیبرید محسوب می شوند اما طراحان# C این مورد را به اندازه ی++ c مهم تلقی نکرده اند. یک زبان هیبرید اجازه ی برنامه نویسی با شیوه های مختلف را میسر می کند. دلیل این که++ C هیبرید است ، این است که قرار بوده تا با زبان C سازگار باشد و همین امر سبب گردیده تا بعضی از جنبه های++ C بسیار پیچیده شوند.

زبان سی شارپ فرض اش بر این است که شما می خواهید تنها برنامه نویسی شیء گرا انجام دهید و همانند++ C مخلوطی از برنامه نویسی رویه ایی (Procedural) و شیء گرا را نمی خواهید به پایان برسانید. بنابراین باید طرز فکر خودتان را با دنیای شیء گرایی تطبیق دهید. در ادامه خواهید دید که در سی شارپ هر چیزی شیء است حتی یک برنامه ی سی شارپ.

در این قسمت با چند دستوررا که لازم داریم را به اختصار توضیح میدهیم.

دستور GET PIXEL :

از این دستور برای گرفتن شدت نور و رنگ پیکسل استفاده می کنیم.

دستور FOR :

از این دستور برای ایجاد حلقه و گرفتن پیکسلها به ترتیب استفاده میکنیم.

دستور BITMAP :

از این دستور برای تعریف یک متغییر از نوع تصویر استفاده می کنیم.

دستور SET PIXEL :

از این دستور برای درج شدت نور و رنگ پیکسل استفاده می کنیم.

فصل دوم

الگوریتمهای پردازش تصویر :

تبدیل یک تصویر رنگی به سیاه و سفید:

برای تبدیل یک تصویررنگی به سیاه و سفید باید شدت نور سه مولفه (RGB) تصویر رنگی را بگیریم و بعد از ان سه مولفه را تقسیم بر سه نماییم. برای پیمایش طول و عرض تصویر از حلقه ها استفاده میکنیم.

الگوریتم:

```
For ( I = 0, I < weith , i++ )
For ( j=0 , j < heigh t ,j++ )
R( I , j ) = ( R(  I . j  ) + G ( I , j ) + B ( I , j  ) ) / 3
```

تبدیل تصویر رنگی یا خاکستری به تصویر سیاه و سفید:

برای تبدیل تصویر رنگی یا خاکستری به سیاه و سفید ابتدا شدت نور پیکسلهای تصویر را می گیریم سپس انرا با مقدار 127 که مینگین شدت نور است مقایسه می کنیم اگر بزرگتر از 127 بود شدت نور ان نقطه را برابر 255 میگیریم و اگر کمتر بود برابر 0 میکنیم.

الگوریتم:

```
For ( I = 0, I < weith , i++ )
For ( j=0 , j < heigh t ,j++ )
If  ( f  ( I ,  j) < 127 )
R ( I , j ) = 0
Else
R ( I , j ) = 255
```

الگوریتم منفی سازی تصویر (negative):

در منفی سازی یک تصویر شدت نور نقاط را با هم عوض می کنیم

تابع منفی سازی تصویر برابر است باs=255-r

الگوریتم:

For (I = 0, I < weith , i++)

For (j=0 , j < heigh t ,j++)

R (I , j) = 255 – f (I , j)

الگوریتم فیلتر های پردازش تصویر:

فیلتر میانگین گیری معمولی(midiling):

اگر در mask تمامی ضرایب w یک باشد رابطه به یک رابطه میانگیری ساده تبدیل خواهد شد:

W1=w2=w3=w4=w5=w7=w8=w9=1

Z5=(z1 + z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9)/9

در واقع میانگین شدت نور 9 پیکسل محاسبه شده و در پیکسل z5 اعمال می شود بدیهی است این روش در حذف نویز موثر است.

فیلتر گاوسی(guse):

الگوریتم:

W1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 = 16

Z5 = (z1 + z2z + z3 + 2z4 + 4z5 + 2z6 + z7 + 2z8 + z9) / 16

فیلتر median :

کاربرد اصلی این فیلتر در حذف نویز فلفل نمکی (salt&prppre)و یک فیلتر خطی و بالا گذر است این فیلتر همسایگیهای z5 را به صورت صعودی یا نزولی مرتب کرده و شدت نور پیکسل وسط را روی عنصر z5 اعمال می کند.

در این فیتر شدت نور پیکسلها را در ارایه ای ذخیره می کنیم سپس مرتب می کنیم.

در برنامه نوشته شده ما سه فیلتر ذکر شده را پیاده سازی می کنیم.

فصل سوم مستدنات برنامه:

```
;using System
;using System.Collections.Generic
;using System.ComponentModel
;using System.Data
;using System.Drawing
;using System.Drawing.Drawing2D
;using System.Text
;using System.Windows.Forms

namespace Transparent_layer
}
public partial class Form1 : Form
}
```

قسمت بالا مربوط به پشته برنامه هست و کدهای نوشته شده را خود c# مي نويسد و نیازی نیست برنامه نویس بنویسد.

```
;Bitmap picture
```

در قسمت روبرو یک متغییر از نوع تصویر تعریف می کنیم و قبل از کد نویسی تعریف می کنیم(یعنی متغییر محلی تعریف می کنیم).

```
()public Form1
}
;()InitializeComponent
{
```

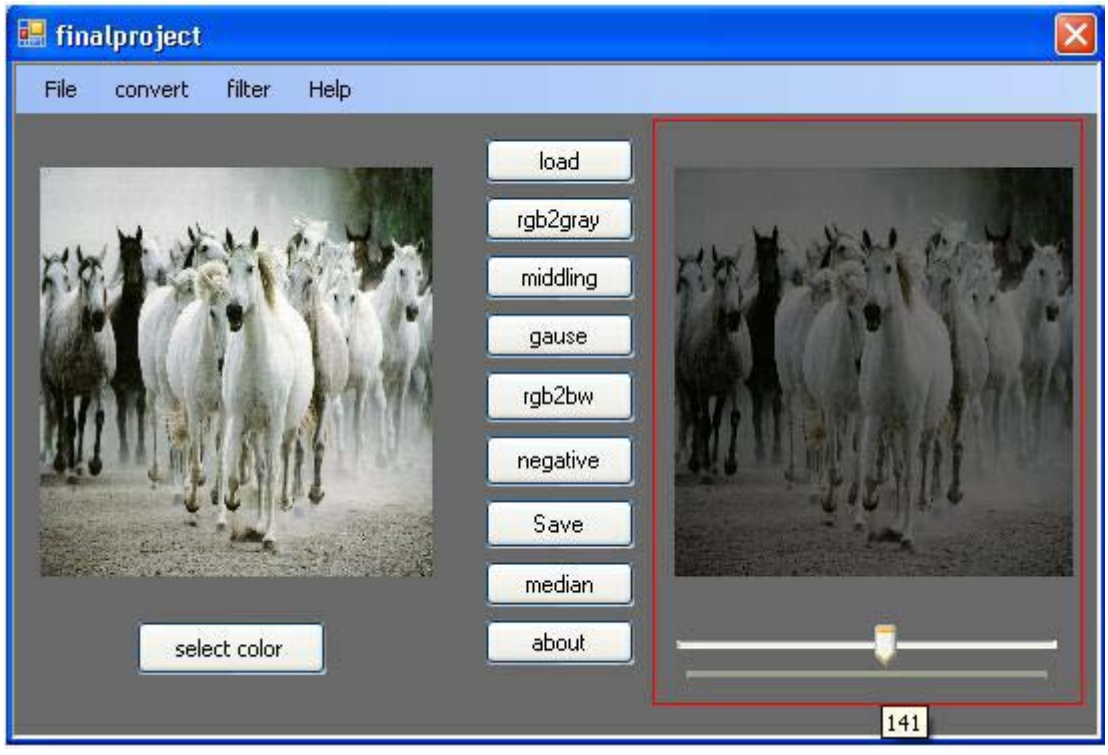
شروع برنامه یا قسمت (code segment) از اینجا شروع مي شود و ما شروع به کد نویسی می کنیم لازم به ذکر است که این قسمت را نیز خود #c ایجا د می کند.

```
private void button1_Click(object sender, EventArgs e)
{
if (colorDialog1.ShowDialog() == DialogResult.OK)
{
```

```
pictureBox1.Image = fill_layer(pictureBox1, pictureBox2.Image, colorDialog1.Color,
trackBar1.Value);
}
```

درقسمت بالا در سطر if مي گوييم اگر پانل انتخاب رنگ خالی نباشد برنامه را ادامه بده و کاربر باید یک رنگ را انتخاب کند کار اصلی این تکه کد انتخاب یک رنگ است و سپس میزان شدت نور انرا با track bar تعیین میکنیم و روي تصوير اعمال میکنیم.

```
private static Image fill_layer(PictureBox Pic_Box, Image img, Color Layer_color, int
Percent)
{
Pic_Box.Image = img;
Bitmap bmp_img = new Bitmap(Pic_Box.Image);
Graphics ghp = Graphics.FromImage(bmp_img);
LinearGradientBrush LineaBrush;
LineaBrush = new LinearGradientBrush(new Rectangle(0, 0, bmp_img.Width,
bmp_img.Height), Color.FromArgb(Percent, Layer_color), Color.FromArgb(Percent,
Layer_color), LinearGradientMode.BackwardDiagonal);
ghp.FillRectangle(LineaBrush, new Rectangle(0, 0, bmp_img.Width, bmp_img.Height));
return (Image)bmp_img;
}
```



کد بالا مربوط به قسمت انتخاب شدت نور یا همان کنتراست تصویر است که در حقیقت برای بالا بردن کنتراست از یک ماسک(بروش)استفاده می کند و تمامی پیکسل ها را پیمایش کرده و شدت نور موردنظر را که توسط track bar که در اول برنامه گفته شد و انتخاب کرده ایم اعمال می کند.

```
private void btnload_Click(object sender, EventArgs e)
{
openFileDialog1.Filter = "jpg file (*.jpg) |*.jpg|"
+ " All files (*.*) |*.*";
openFileDialog1.FilterIndex = 1;
openFileDialog1.Title = "load pic";
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
picture = new Bitmap(openFileDialog1.FileName);

pictureBox2.Image = picture;
}
```

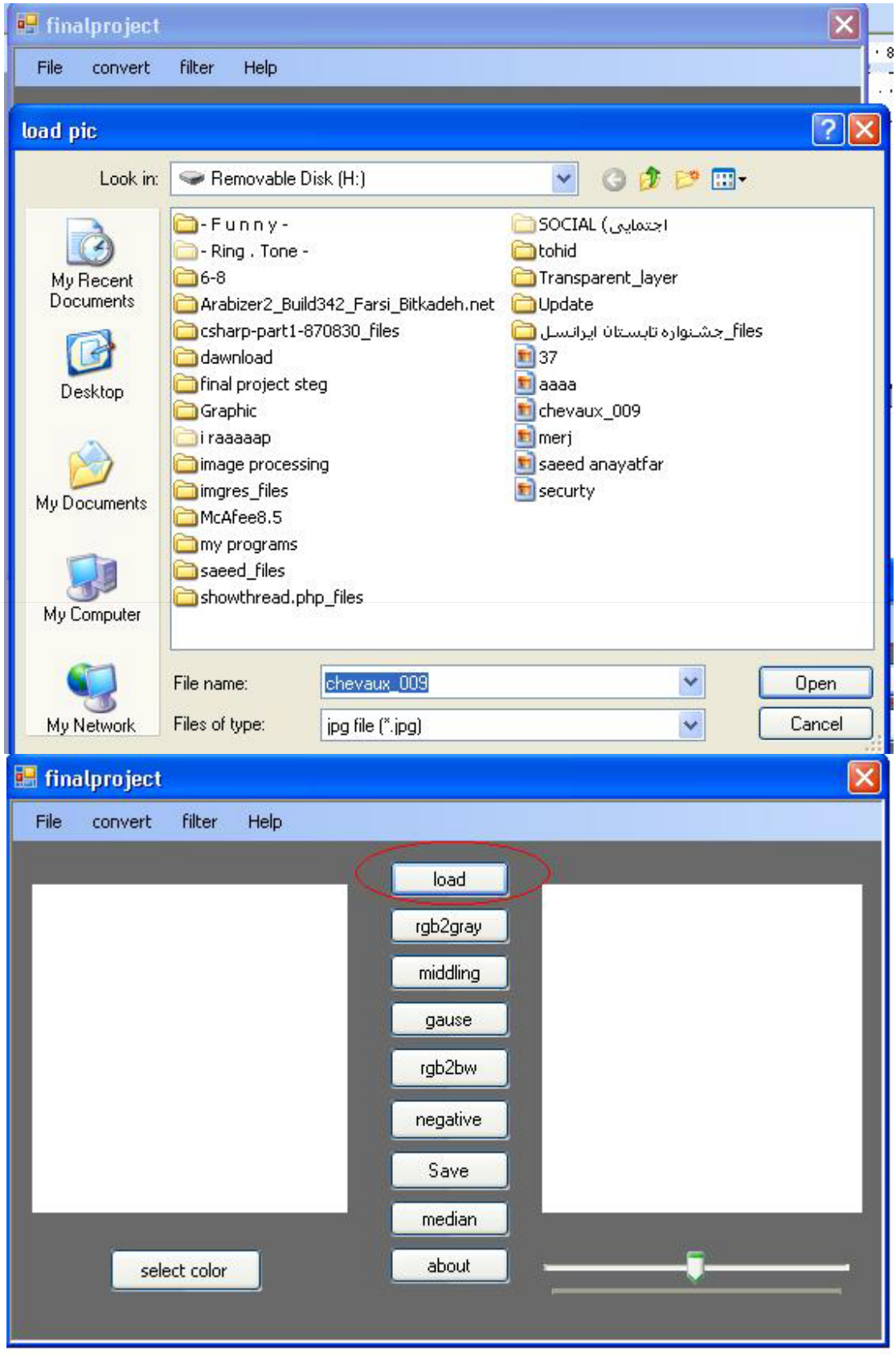
این تکه کد یک عکس را برای ما لود میکند در سطر دوم این برنامه ما یک فیلتر میگذاریم و می کوییم فقط فایل هایی را باز کند که پسوند انها مربوط به فرمت تصویر است .

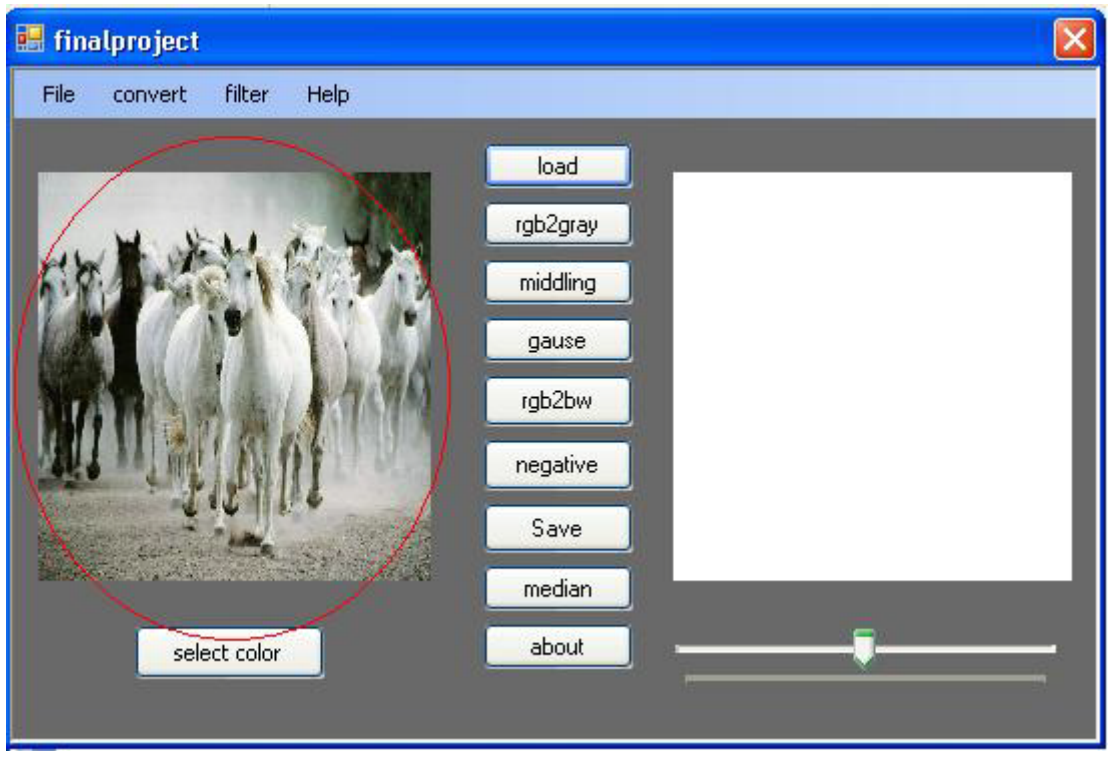
در سطر پنجم عنوان پنجره ای که باز می شود را می نویسیم.

در سطر بعدی می گوییم که کاربر یک فایل انتخاب کند و اگر نکند برنامه به سطر بعدی نرود.

در سطر هفتم متغییر picture را که در بالا تعریف نموده ایم مقدار دهی می کنیم لازم به ذکر است قبل از استفاده از متغییر باید new را تایپ کنیم ما در متغییر که تعریف کرده ایم ادرس فایل را قرار می دهیم.

در سطر اخردر کادر نمایش تصویر که قبلا ایجاد کردهایم تصویر لود شده را نمایش می دهیم.






```
private void btnrgb2gray_Click(object sender, EventArgs e)
{
int i, j, r1, r2, r3, z;
Color k;
for (i = 0; i < picture.Width; i++)
for (j = 0; j < picture.Height; j++)
{
k = picture.GetPixel(i, j);
r1 = k.R;
r2 = k.G;
r3 = k.B;
z = (r1 + r2 + r3) / 3;
picture.SetPixel(i, j, Color.FromArgb(z, z, z));
}
pictureBox1.Image = picture;
}
```

قسمت بالا مربوط به دکمه تبدیل تصویر رنگی به سیاه و سفید است در سطراول کد مربوط به رویداد نوشته شده است .

در سطر دوم 7 متغییر از نوع عدد صحیح تعریف شده است.

در سطر سوم یک متغییر از نوع رنگ تعریف میکنیم تا رنگ تصویر را در ان نگهداری بکنیم لازم به ذکر است متغییر از نوع رنگ دارای سه مولفه قرمز.ابی و سیاه است.

در سطر جهارم و پنجم توسط دو حلقه ی تودرتو پیکسل های عرض و طول تصویر را پیمایش می کنیم.از یک حلقه برای پیمایش عرض و از دیگری برای پیمایش طول تصویر استفاده می کنیم.

در سطر ششم شدت نور و میزان اشباع رنگ تصویر را می گیریم و در متغییر قرار می دهیم.

در سطر هفتم مولفه ی رنگ قرمز تصویر را در متغییر ذخیره می کنیم.

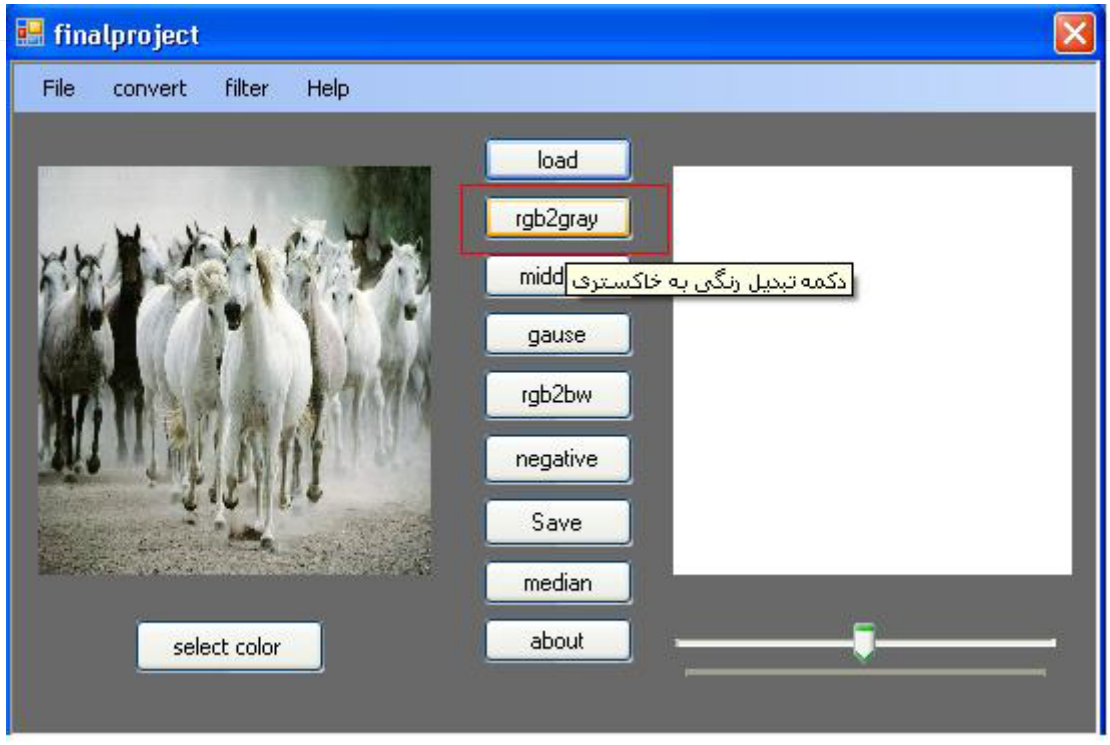
در سطر هشتم مولفه ی سبز قرمز تصویر را در متغییر ذخیره می کنیم.

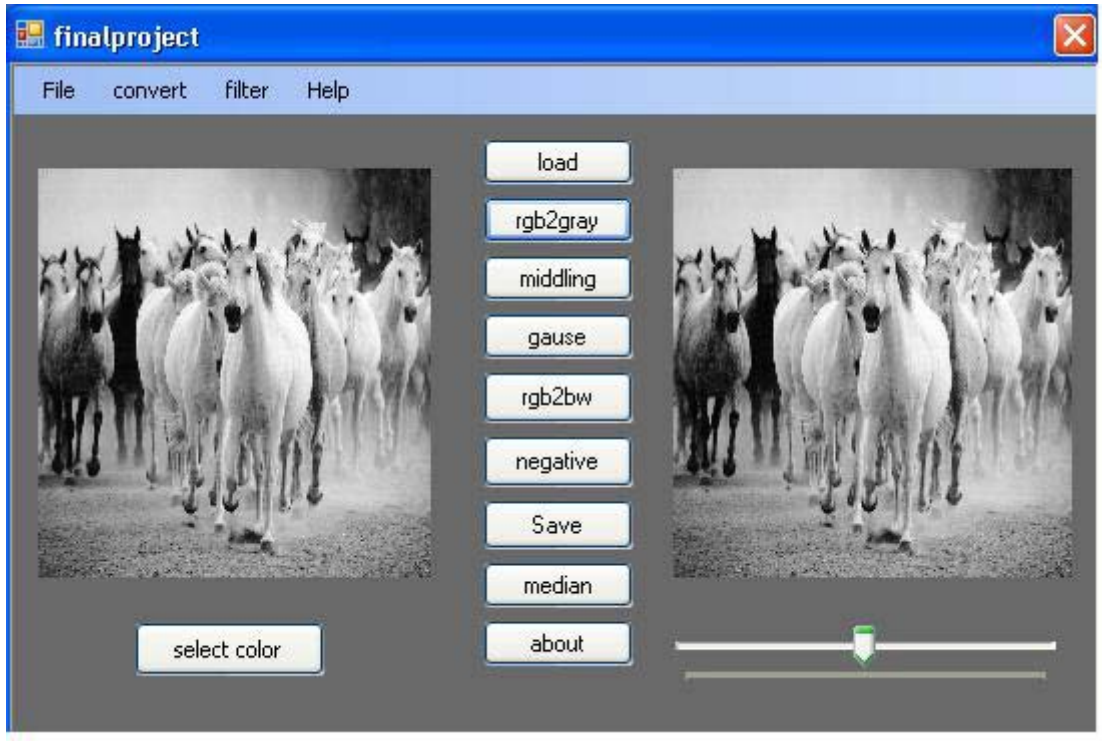
در سطر نهم مولفه ی رنگ سیاه تصویر را در متغییر ذخیره می کنیم.

در سطر دهم میزان شدت نور سه مولفه ی تصویر را طبق الگوریتم تبدیل رنگی به سیاه و سفید باهم جمع می کنیم و سپس تقسیم بر سه می نماییم و در یک متغییر دیگه که در بالا تعریف نموده ایم قرار می دهیم.

در سطر یازدهم تصویر را با شدت نور بدست امده از الگوریتم دوباره پیکسل به پیکیل می نویسم.

در نهایت در سطر اخر تصویر بدست امده را در کادر نمایش تصویر نمایش می دهیم.



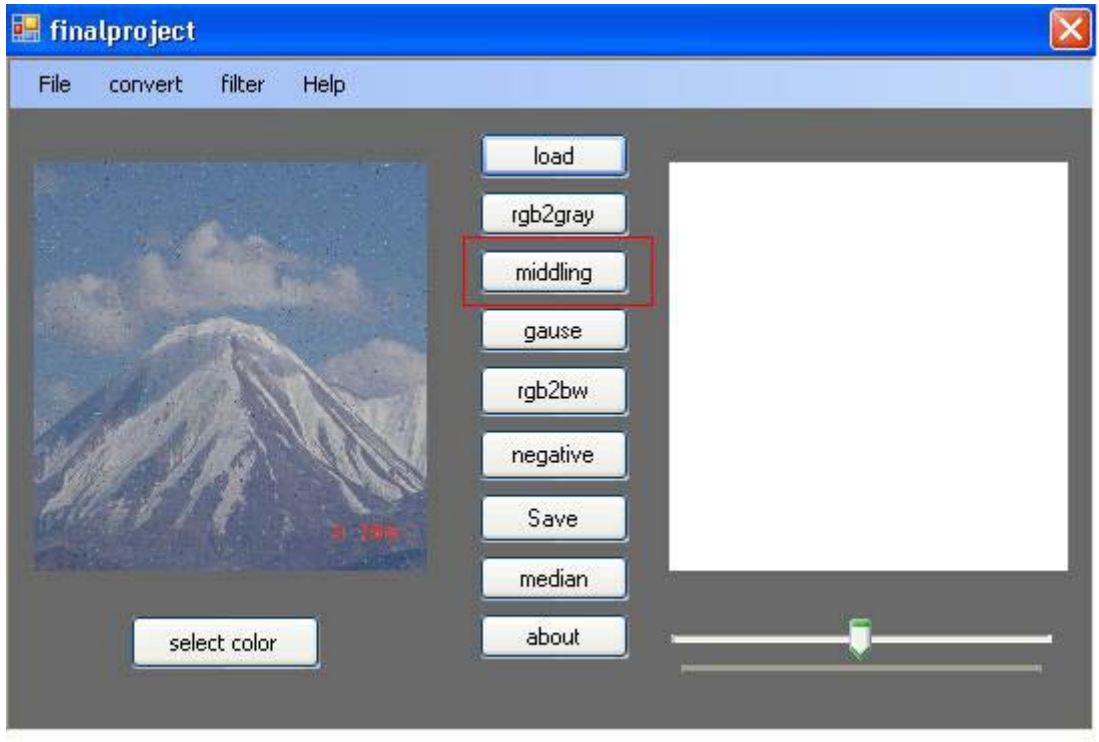


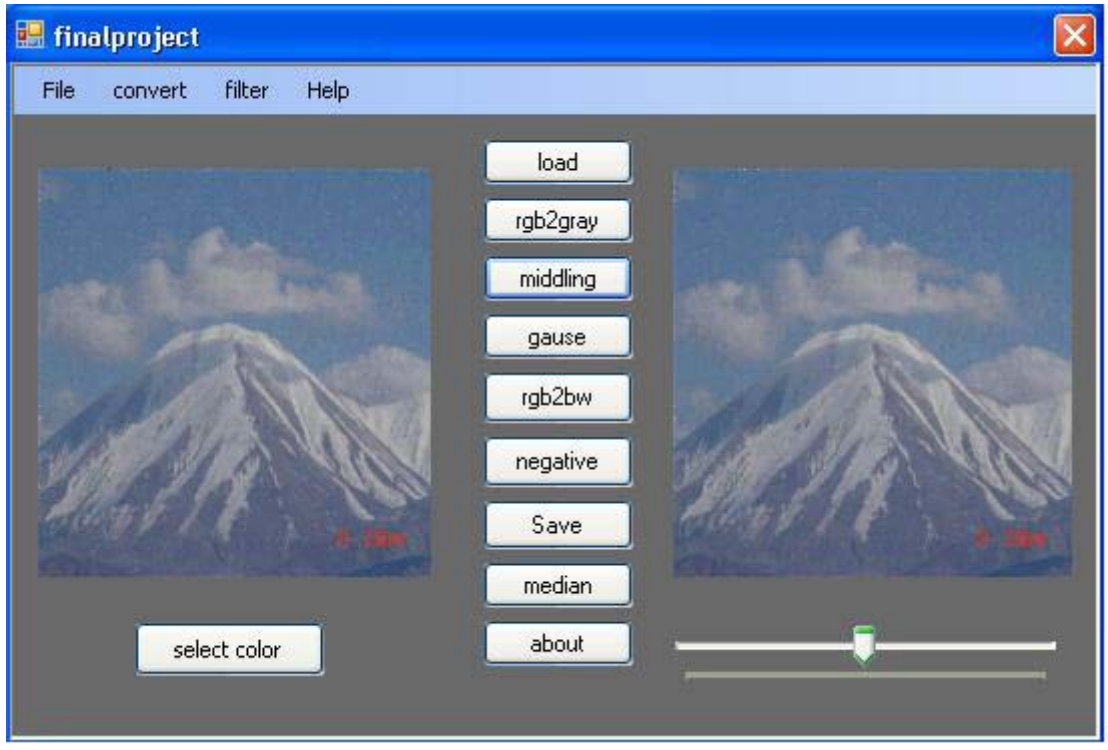

```
private void btnmedian_Click(object sender, EventArgs e)
{
int i, j;
int zr, zg, zb;
Color c1, c2, c3, c4, c5, c6, c7, c8, c9;
for (i = 2; i <= picture.Width - 2; i++)
for (j = 2; j < picture.Height - 2; j++)
{
c1 = picture.GetPixel(i - 1, j - 1);
c2 = picture.GetPixel(i, j - 1);
c3 = picture.GetPixel(i + 1, j - 1);
c4 = picture.GetPixel(i - 1, j);
c5 = picture.GetPixel(i, j);
c6 = picture.GetPixel(i + 1, j);
c7 = picture.GetPixel(i - 1, j + 1);
c8 = picture.GetPixel(i, j + 1);
```

```
c9 = picture.GetPixel(i + 1, j + 1);

zr = (c1.R / 9 + c2.R / 9 + c3.R / 9 + c4.R / 9 + c5.R / 9 + c6.R / 9 + c7.R / 9 + c8.R / 9 +
c9.R / 9);
zg = (c1.G / 9 + c2.G / 9 + c3.G / 9 + c4.G / 9 + c5.G / 9 + c6.G / 9 + c7.G / 9 + c8.G / 9
+ c9.G / 9);
zb = (c1.B / 9 + c2.B / 9 + c3.B / 9 + c4.B / 9 + c5.B / 9 + c6.B / 9 + c7.B / 9 + c8.B / 9 +
c9.B / 9);

picture.SetPixel(i, j, Color.FromArgb(zr, zg, zb));
}

}
pictureBox1.Image = picture;
```

تکه کد بالا مربوط به فیلتر میانگین گیری است در این کد ما شدت نور خانه های اطراف پیکسل نویز دار را می گیریم سپس باهم جمع می کنیم و در نهایت تقسیم بر تعداد خانه ها می کنیم.

این فیتر برای نواحی ارام تصویر بسیار موثر است.



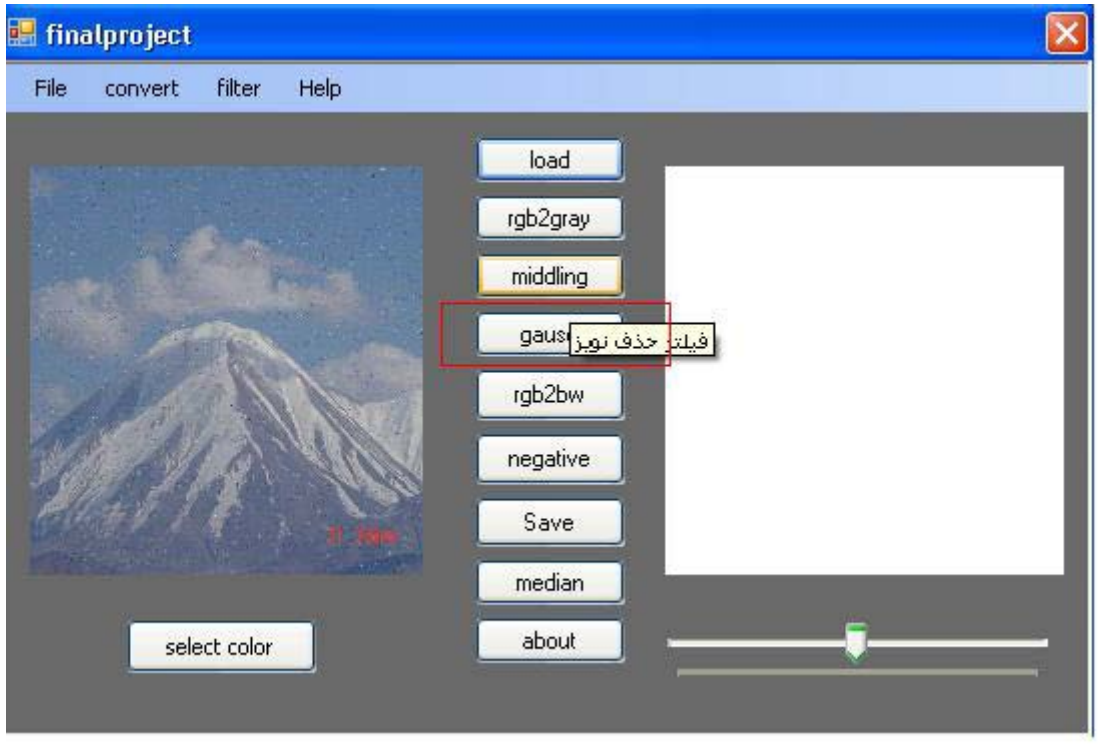


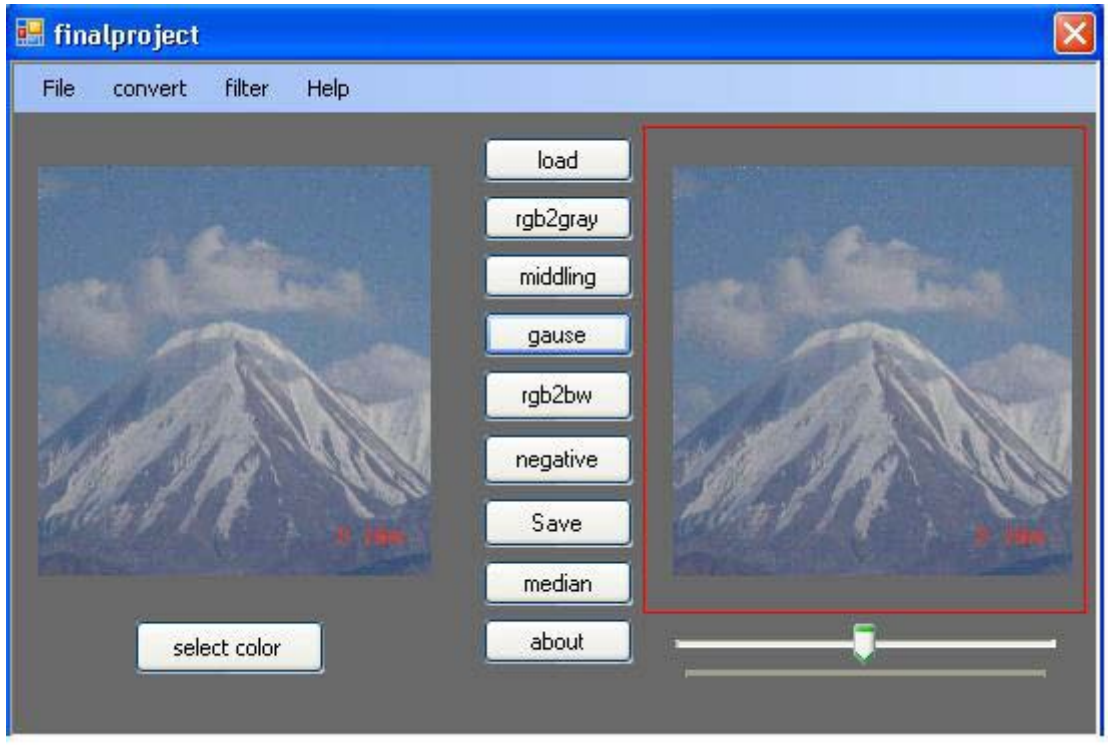

```
private void btngause_Click(object sender, EventArgs e)
{
int i, j;
Color z1, z2, z3, z4, z5, z6, z7, z8, z9;
int z5r, z5g, z5b;
for (i = 2; i < picture.Width - 2; i++)
for (j = 2; j < picture.Height - 2; j++)
{
z1 = picture.GetPixel(i - 1, j - 1);
z2 = picture.GetPixel(i, j - 1);
z3 = picture.GetPixel(i + 1, j - 1);
z4 = picture.GetPixel(i - 1, j);
z5 = picture.GetPixel(i, j);
z6 = picture.GetPixel(i + 1, j);
z7 = picture.GetPixel(i - 1, j + 1);
z8 = picture.GetPixel(i, j + 1);
z9 = picture.GetPixel(i + 1, j + 1);
```

```
z5r = (z1.R + 2 * z2.R + z3.R + 2 * z4.R + 4 * z5.R + 2 * z6.R + z7.R + 2 * z8.R + z9.R)
/ 16;
z5g = (z1.G + 2 * z2.G + z3.G + 2 * z4.G + 4 * z5.G + 2 * z6.G + z7.G + 2 * z8.G +
z9.G) / 16;
z5b = (z1.B + 2 * z2.B + z3.B + 2 * z4.B + 4 * z5.B + 2 * z6.B + z7.B + 2 * z8.B + z9.B)
/ 16;

picture.SetPixel(i, j, Color.FromArgb(z5r, z5g, z5b));
pictureBox1.Image = picture;
}
}
```

کدهای بالا مربوط به فیتر گاوسی است.

فیلتر گاوسی خانه هایی که با پیکسل مورد نظر تقارن طولی و عرضی دارند را گرفته و شدت نور انها را گرفته و باهم جمع می نماید لازم به ذکر است خود خانه ای که دارای نویز هست را نمیگیرد و همچنین در تقسیم نیز انرا حساب نمی کند.






```
private void btngrayetobw_Click(object sender, EventArgs e)
{
int i, j, r1, r2, r3, z, r;
Color k;
for (i = 0; i < picture.Width; i++)
for (j = 0; j < picture.Height; j++)
{
r = 127;

k = picture.GetPixel(i, j);
r1 = k.R;



r2 = k.G;
r3 = k.B;
z = (r1 + r2 + r3) / 3;
if (z < r)
```

```
z = 0;
else
z = 255;
picture.SetPixel(i, j, Color.FromArgb(z, z, z));

}
pictureBox1.Image = picture;
}

private void btnsave_Click(object sender, EventArgs e)
{
saveFileDialog1.ShowDialog();
picture.Save(saveFileDialog1.FileName);
}
```

کد بالا یک تصئیر را می گیرد سپس انرا به سیاه و سفید تبدیل می کند.

در سطر یک و دو به ترتیب متغییر هایی از نوع عدد صحیح و رنگ تعریف می کنیم.

در سطر سوم و جهارم از دو حلقه ی تودرتو برای گرفتن پیکسل ها ی طول و عرض تصویر استفاده می کنیم .

در سطر پنجم یکی از متغییر ها را با عدد 127 مقدار دهی میکنیم و برای این عدد 127 را میگیریم که شدت نور همیشه عددی بین 0و255 است و ما نصف انرا در نظر می گیریم.

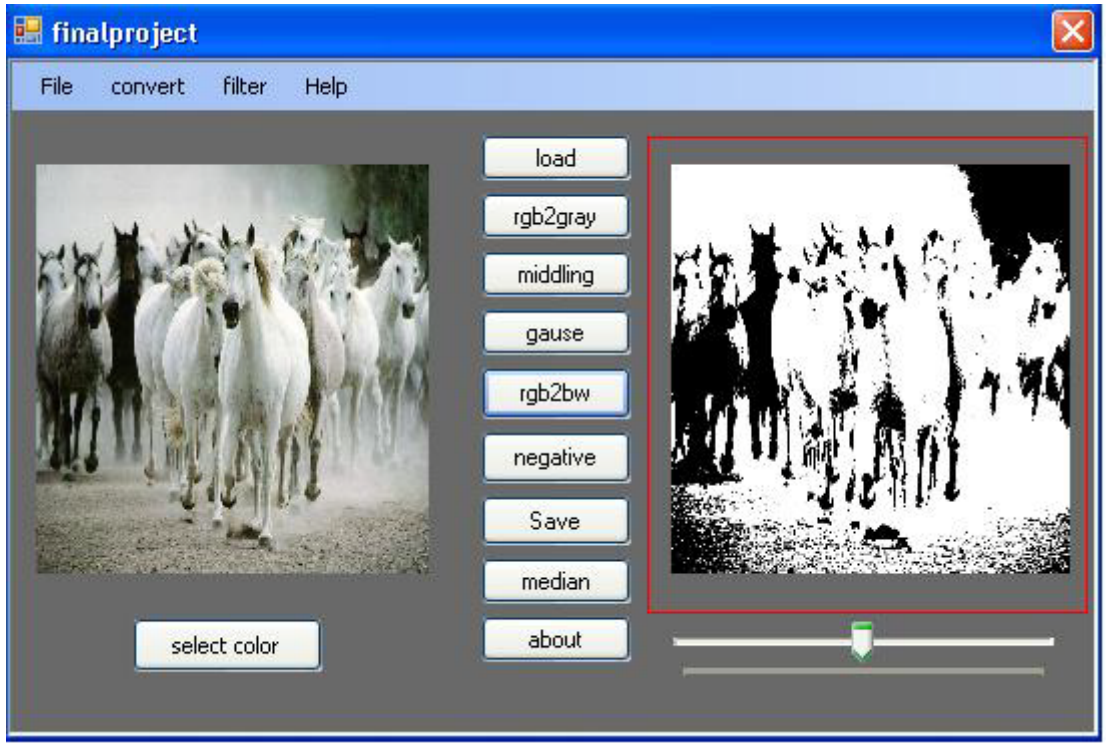
در سطر ششم شدت نور پیکسلا ها را دریافت می کنیم و در یک متغیر از نوع رنگ که در بالا تعریف نمودهایم نگهداری میکنیم .

در سطر های 7 تا 9 سه مولفه ی رنگ تصویر را می گیریم.

در سطر 10 سه تا مولفه را جمع و تقسیم بر سه مینماییم.

در سطر بعدی توسط یک شرط می گوییم اگر شدت نور بدست امده از 127 کوچکتر اس یا نه . اگر کوچتکر باشد انرا برابر 0 و اگر بزرگتر بود برابر 255 قرارر می دهیم.

لازم به ذکر است تصویر سیاه و سفید دارای دو نوع شدت نوراست یا 0 یا 255 .



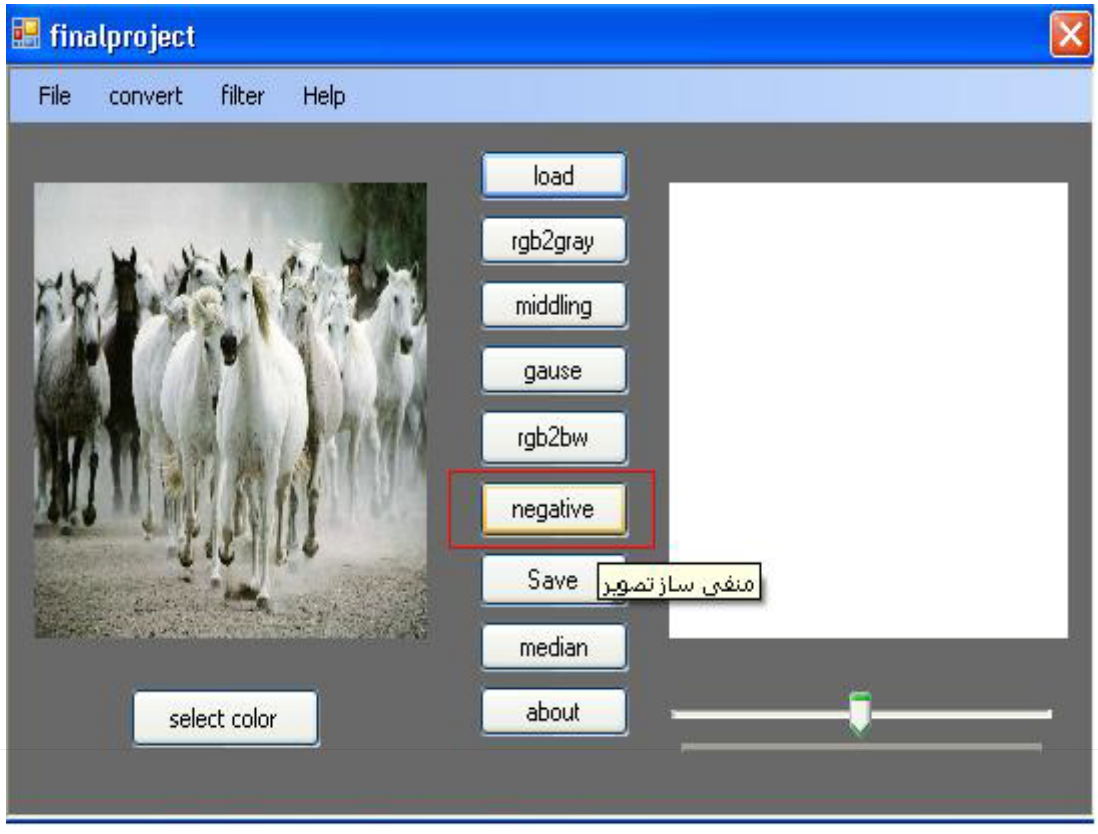

```
private void btnnegative_Click(object sender, EventArgs e)
{
int i, j, r1, r2, r3;
Color k;
for (i = 0; i < picture.Width; i++)
for (j = 0; j < picture.Height; j++)
{
k = picture.GetPixel(i, j);
r1 = 255 - k.R;
r2 = 255 - k.G;
r3 = 255 - k.B;
picture.SetPixel(i, j, Color.FromArgb(r1, r2, r3));
}
pictureBox1.Image = picture;
}
```

در کد بالا ما یک تصویر را میگیریم و منفی انرا نمایش می دهیم.

در تکه کد بالا طبق روال نمدنه های قبل در ابتدا شدت نور تصویر را می گیریم.

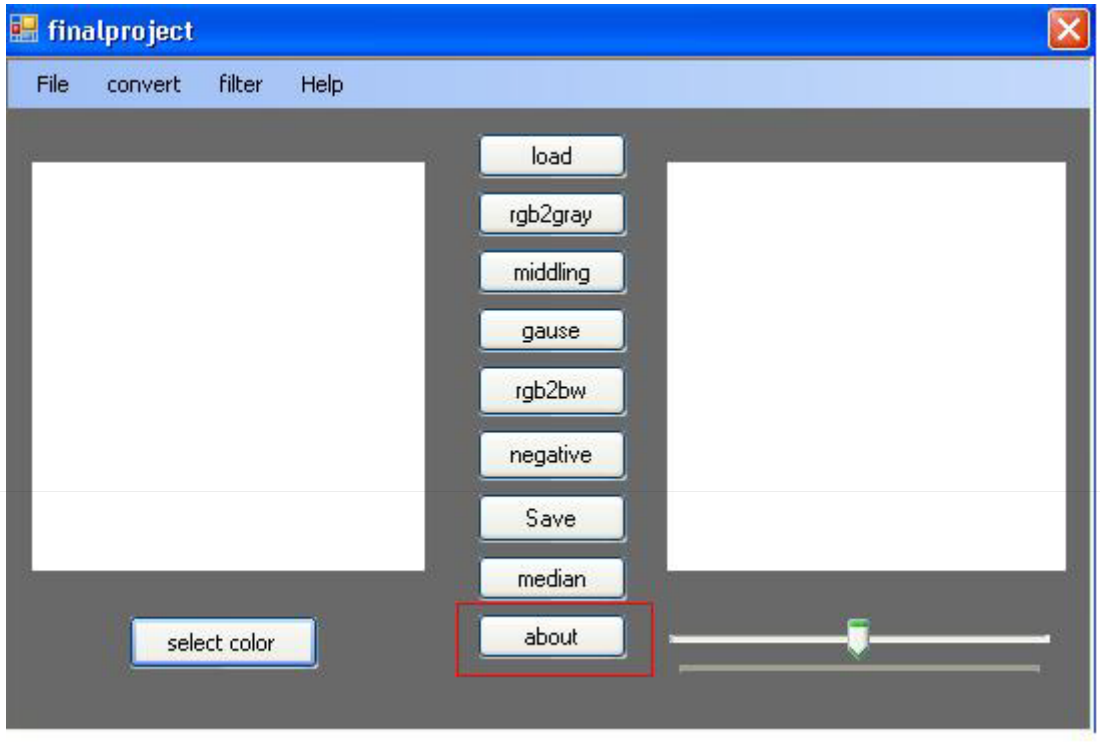
شدت نور حاصله را از عدد 255 کم می نماییم طبق الگریتم گفته شده .

در سطر اخر تصویر را با شدت نور های بدست امده دوباره می نویسیم.



```
public void ShowAboutBox()

{

AboutBox1 objAbout = new AboutBox1();
objAbout.ShowDialog(this);
}
```

تکه کد بالا مشخصات اسمبلی و نسخه و نویسنده برنامه را نمایش میدهد.




```
private void btnmedian_Click_1(object sender, EventArgs e)
{
int i, j;
Color c1, c2, c3, c4, c5, c6, c7, c8, c9;
for (i = 2; i <= picture.Width - 2; i++)
for (j = 2; j < picture.Height - 2; j++)
{
c1 = picture.GetPixel(i - 1, j - 1);
c2 = picture.GetPixel(i, j - 1);
```

```
c3 = picture.GetPixel(i + 1, j - 1);
c4 = picture.GetPixel(i - 1, j);
c5 = picture.GetPixel(i, j);
c6 = picture.GetPixel(i + 1, j);
c7 = picture.GetPixel(i - 1, j + 1);
c8 = picture.GetPixel(i, j + 1);
c9 = picture.GetPixel(i + 1, j + 1);
int[] zr = new int[9] { c1.R, c2.R, c3.R, c4.R, c5.R, c6.R, c7.R, c8.R, c9.R };
int[] zg = new int[9] { c1.G, c2.G, c3.G, c4.G, c5.G, c6.G, c7.G, c8.G, c9.G };
int[] zb = new int[9] { c1.B, c2.B, c3.B, c4.B, c5.B, c6.B, c7.B, c8.B, c9.B };



Array.Sort(zr);
Array.Sort(zb);
Array.Sort(zg);




picture.SetPixel(i, j, Color.FromArgb(zr[5], zg[5], zb[5]));


}
pictureBox1.Image = picture;
}
```

در قسمت بالا کد فیلتر میانه نوشته شده است.

در سطرهای 1 تا 4 طبق برنامه های قبل متغییرهایی از نوع صحیح و رنک تعریف می کنیم و پیکسل های تصویر را می گیریم.

در سطر های 5 تا 10 پیکسل های اطراف پیکسل خراب را می گیریم.

در سطر بعد یک ارایه تعریف می کنیم و شدت نور های مربوط به رنگ قرمز را در ان نگهداری می کنیم.

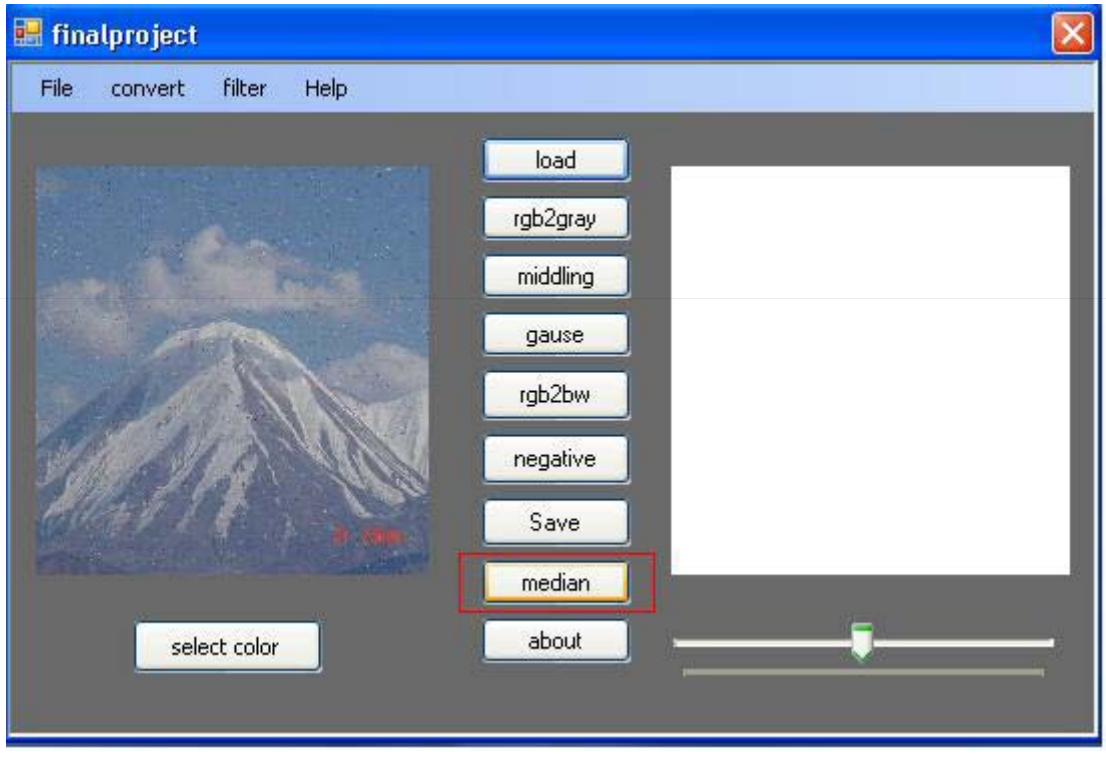
در سطر بعد یک ارایه تعریف می کنیم و شدت نور های مربوط به رنگ سبز را در ان نگهداری می کنیم.

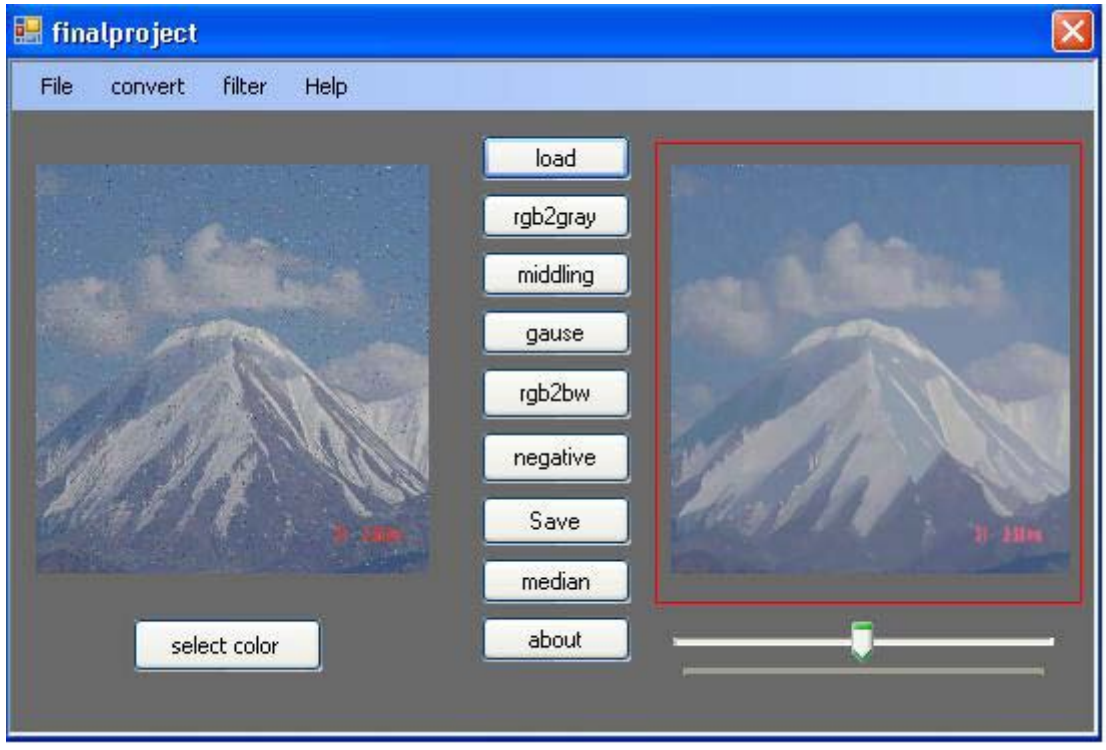
در سطر بعد یک ارایه تعریف می کنیم و شدت نور های مربوط به رنگ سیاه را در ان نگهداری می کنیم.

در سه سطر بعدی ارایه های ایجاد شده را مرتب می کنیم.

در سطر بعدی تصویر را با شدت نورهای جدید می نویسیم و شدت نور خانه ی پنجم ارایه را به جای شدت نور خراب قرار می دهیم.

در سطر بعدی تصویر حاصل از فیلتر را نمایش می دهیم.






```
private void openToolStripMenuItem1_Click(object sender, EventArgs e)
{
openFileDialog1.Filter = "jpg file (*.jpg) |*.jpg|"
+ " All files (*.*) |*.*";
openFileDialog1.FilterIndex = 1;
openFileDialog1.Title = "load pic";
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
picture = new Bitmap(openFileDialog1.FileName);
pictureBox2.Image = picture;
}



}

private void exitToolStripMenuItem1_Click(object sender, EventArgs e)
{
```

```
Application.Exit();
}

private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
{
AboutBox1 objAbout = new AboutBox1();
objAbout.ShowDialog(this);
}

private void saveToolStripMenuItem1_Click(object sender, EventArgs e)
{
saveFileDialog1.ShowDialog();
picture.Save(saveFileDialog1.FileName);
}

private void printToolStripMenuItem_Click(object sender, EventArgs e)
{

}

private void undoToolStripMenuItem_Click(object sender, EventArgs e)
{
}

private void customizeToolStripMenuItem_Click(object sender, EventArgs e)
{
int i, j, r1, r2, r3, z;
Color k;
for (i = 0; i < picture.Width; i++)
for (j = 0; j < picture.Height; j++)
```

```
{
k = picture.GetPixel(i, j);
r1 = k.R;
r2 = k.G;
r3 = k.B;
z = (r1 + r2 + r3) / 3;
picture.SetPixel(i, j, Color.FromArgb(z, z, z));
}
pictureBox1.Image = picture;
}

private void optionsToolStripMenuItem_Click(object sender, EventArgs e)
{
int i, j, r1, r2, r3, z, r;
Color k;
for (i = 0; i < picture.Width; i++)
for (j = 0; j < picture.Height; j++)
{
r = 127;

k = picture.GetPixel(i, j);
r1 = k.R;



r2 = k.G;
r3 = k.B;
z = (r1 + r2 + r3) / 3;
if (z < r)
z = 0;
else
```

```
z = 255;
picture.SetPixel(i, j, Color.FromArgb(z, z, z));
}
}

private void negativeToolStripMenuItem_Click(object sender, EventArgs e)
{
int i, j, r1, r2, r3;
Color k;
for (i = 0; i < picture.Width; i++)
for (j = 0; j < picture.Height; j++)
{
k = picture.GetPixel(i, j);
r1 = 255 - k.R;
r2 = 255 - k.G;
r3 = 255 - k.B;

picture.SetPixel(i, j, Color.FromArgb(r1, r2, r3));
}
pictureBox1.Image = picture;
}

private void midilingToolStripMenuItem_Click(object sender, EventArgs e)
{
int i, j;
int zr, zg, zb;
Color c1, c2, c3, c4, c5, c6, c7, c8, c9;
for (i = 2; i <= picture.Width - 2; i++)
for (j = 2; j < picture.Height - 2; j++)
{
```

```
c1 = picture.GetPixel(i - 1, j - 1);
c2 = picture.GetPixel(i, j - 1);
c3 = picture.GetPixel(i + 1, j - 1);
c4 = picture.GetPixel(i - 1, j);
c5 = picture.GetPixel(i, j);
c6 = picture.GetPixel(i + 1, j);
c7 = picture.GetPixel(i - 1, j + 1);
c8 = picture.GetPixel(i, j + 1);
c9 = picture.GetPixel(i + 1, j + 1);

zr = (c1.R / 9 + c2.R / 9 + c3.R / 9 + c4.R / 9 + c5.R / 9 + c6.R / 9 + c7.R / 9 + c8.R / 9 +
c9.R / 9);
zg = (c1.G / 9 + c2.G / 9 + c3.G / 9 + c4.G / 9 + c5.G / 9 + c6.G / 9 + c7.G / 9 + c8.G / 9
+ c9.G / 9);
zb = (c1.B / 9 + c2.B / 9 + c3.B / 9 + c4.B / 9 + c5.B / 9 + c6.B / 9 + c7.B / 9 + c8.B / 9 +
c9.B / 9);

picture.SetPixel(i, j, Color.FromArgb(zr, zg, zb));

}
pictureBox1.Image = picture;
}

private void gauseToolStripMenuItem_Click(object sender, EventArgs e)
{
int i, j;
Color z1, z2, z3, z4, z5, z6, z7, z8, z9;
int z5r, z5g, z5b;
for (i = 2; i < picture.Width - 2; i++)
for (j = 2; j < picture.Height - 2; j++)
```

```
{
z1 = picture.GetPixel(i - 1, j - 1);
z2 = picture.GetPixel(i, j - 1);
z3 = picture.GetPixel(i + 1, j - 1);
z4 = picture.GetPixel(i - 1, j);
z5 = picture.GetPixel(i, j);
z6 = picture.GetPixel(i + 1, j);
z7 = picture.GetPixel(i - 1, j + 1);
z8 = picture.GetPixel(i, j + 1);
z9 = picture.GetPixel(i + 1, j + 1);

z5r = (z1.R + 2 * z2.R + z3.R + 2 * z4.R + 4 * z5.R + 2 * z6.R + z7.R + 2 * z8.R + z9.R)
/ 16;
z5g = (z1.G + 2 * z2.G + z3.G + 2 * z4.G + 4 * z5.G + 2 * z6.G + z7.G + 2 * z8.G +
z9.G) / 16;
z5b = (z1.B + 2 * z2.B + z3.B + 2 * z4.B + 4 * z5.B + 2 * z6.B + z7.B + 2 * z8.B + z9.B)
/ 16;

picture.SetPixel(i, j, Color.FromArgb(z5r, z5g, z5b));
pictureBox1.Image = picture;
}
}

private void medianToolStripMenuItem_Click(object sender, EventArgs e)
{
int i, j;



Color c1, c2, c3, c4, c5, c6, c7, c8, c9;
for (i = 2; i <= picture.Width - 2; i++)
```

```
for (j = 2; j < picture.Height - 2; j++)
{
c1 = picture.GetPixel(i - 1, j - 1);
c2 = picture.GetPixel(i, j - 1);
c3 = picture.GetPixel(i + 1, j - 1);
c4 = picture.GetPixel(i - 1, j);
c5 = picture.GetPixel(i, j);
c6 = picture.GetPixel(i + 1, j);
c7 = picture.GetPixel(i - 1, j + 1);
c8 = picture.GetPixel(i, j + 1);
c9 = picture.GetPixel(i + 1, j + 1);
int[] zr = new int[9] { c1.R, c2.R, c3.R, c4.R, c5.R, c6.R, c7.R, c8.R, c9.R };
int[] zg = new int[9] { c1.G, c2.G, c3.G, c4.G, c5.G, c6.G, c7.G, c8.G, c9.G };
int[] zb = new int[9] { c1.B, c2.B, c3.B, c4.B, c5.B, c6.B, c7.B, c8.B, c9.B };
```

```
Array.Sort(zr);
Array.Sort(zb);
Array.Sort(zg);
picture.SetPixel(i, j, Color.FromArgb(zr[5], zg[5], zb[5]));

}
pictureBox1.Image = picture;
}

private void Form1_Load(object sender, EventArgs e)
{

}
}
```

}

در سطر های بالا کلیه دستورات نوشته شده در برنامه اصلی را در منو کپی می کنیم و با این عمل میتوانیم از منو ها نیز استفاده کنیم.

فصل چهارم:

نتايج:

اولين قدم در بسياري از الگوريتم هاي پردازش تصوير و بينايي ماشين، حذف نويز از تصوير مي باشد، چرا كه بدون حذف نويز اين الگوريتم ها نتايج خوبي توليد نمي كنند . براي مثال نويز نقطه اي كه به مانند لكه هاي سياه يا روشن روي تصوير مي باشد يكي از انواع شايع نويزها مي باشد . تا به حال فيلترهاي مختلفي براي كاهش اين نوع نويز ارائه شده است . در گروه اول، فيلترهايي ارائه شد كه عملكرد خوبي در حذف نويز داشته ولي عموماً تصوير را مات و جزئيات آن را از بين مي بردند . گروه دوم فيلترهايي كه ارائه شد، جزئيات تصوير را حفظ كرده ولي ميزان كاهش نويز آنها كم بود و گروه سوم فيلترها، آنهايي هستند كه تلاش مي كنند نويز را حذف كرده در حالي كه لبه ها و جزئيات تصوير را نيز حفظ كنند . در بسياري از موارد اين فيلترها عملكرد خوبي دارد ولي چنانچه جزئيات ريز تصوير كم باشد و يا سطوح خاكستري موجود در همسايگي پيكسل ها اختلاف زيادي با هم داشته باشند، باعث مات شدن تصوير مي گردد.

دوربین های دیجیتالی، امروزه با گسترش روز افزون روش های مختلف اخذ اطلاعات گسسته مانند پویشگرها و اطلاعات همواره کم و بیش همراه مقداری پردازش تصویر کاربرد فراوانی یافته است. تصاویر حاصله از این محوشدگی مرزهای نمونه های داخل تصویر می باشند که موجب کاهش نویز بوده و در مواردی نیز دارای مشکل گردند. مجموعه عملیات و روش هایی که به منظور کاهش عیوب و افزایش کیفیت وضوح تصویر دریافتی می تصویر مورد استفاده قرار میگیرد، پردازش تصویر نامیده می شود. اگرچه حوزه های کار با ظاهری تصویر ، (Enhancement)ظاهری بسیار وسیع است ولی عموماً محدوده مورد توجه در چهار زمینه ی بهبود کیفیت (Compression and Coding) تصویر ، فشرده گی و رمزگذاری(Restoration)بازسازی تصاویر مختل شده گردد متمرکز می (Understanding) و درک تصویر توسط ماشین.

کیفیت دیداری فیلتر محو کننده و افزایش تضاد برای بهتر کردن بهبود تصاویر شامل روش هایی مثل استفاده از به روش تصاویر و اطمینان از نمایش درست آن ها در محیط مقصد است. بینایی ماشین تصاویر را درک کرد تا هایی می پردازد که به کمک آن ها می توان معنی و محتوای از آن ها در کارهایی چون رباتیک و محور تصاویر استفاده شود .

فهرست منابع و ماخذ:

قوانین انجام پروژه ها:
نوشته خانم زهرا کریمی

آموزش #c:
نوشته سید محمد هاشمیان

اصول اولیه پردازش تصویر:
نوشته مرتضی سر گلزایی جوان